

配置 Babel

你或许已经注意到了，目前为止通过运行 Babel 自己我们并没能“翻译”代码，而仅仅是把代码从一处拷贝到了另一处。

这是因为我们还没告诉 Babel 要做什么。

由于 Babel 是一个可以用各种花样去使用的通用编译器，因此默认情况下它反而什么都不做。你必须明确地告诉 Babel 应该要做什么。

你可以通过安装**插件（plugins）**或**预设（presets，也就是一组插件）**来指示 Babel 去做什么事情。（所谓的 presets 其实就是一些同类 plugin 打包的结果，方便进行添加）

.babelrc

在我们告诉 Babel 该做什么之前，我们需要创建一个配置文件。你需要做的就是项目的根路径下创建 .babelrc 文件。然后输入以下内容作为开始：

```
{
  "presets": [],
  "plugins": []
}
```

这个文件就是用来让 Babel 做你要它做的事情的配置文件。

注意：尽管你也可以用其他方式给 Babel 传递选项，但 .babelrc 文件是约定也是最好的方式。

babel-preset-es2015

我们先从让 Babel 把 ES2015（最新版本的 JavaScript 标准，也叫做 ES6）编译成 ES5（现今在大多数 JavaScript 环境下可用的版本）开始吧。

我们需要安装 "es2015" Babel 预设：

```
$ npm install --save-dev babel-preset-es2015
```

我们修改 .babelrc 来包含这个预设。

```
{
  "presets": [
+   "es2015"
  ],
  "plugins": []
}
```

babel-preset-react

设置 React 一样容易。只需要安装这个预设：

```
$ npm install --save-dev babel-preset-react
```

然后在 .babelrc 文件里补充：

```
{
```

```
"presets": [  
  "es2015",  
+  "react"  
],  
"plugins": []  
}
```

babel-preset-stage-x

JavaScript 还有一些提案，正在积极通过 TC39（ECMAScript 标准背后的技术委员会）的流程成为标准的一部分。

这个流程分为 5（0—4）个阶段。随着提案得到越多的关注就越有可能被标准采纳，于是他们就继续通过各个阶段，最终在阶段 4 被标准正式采纳。

以下是 4 个不同阶段的（打包的）预设：

- babel-preset-stage-0
- babel-preset-stage-1
- babel-preset-stage-2
- babel-preset-stage-3

注意 stage-4 预设是不存在的因为它就是上面的 es2015 预设。

以上每种预设都依赖于紧随的后期阶段预设。例如，babel-preset-stage-1 依赖 babel-preset-stage-2，后者又依赖 babel-preset-stage-3。

Stage 0:

- Function Bind Syntax: 函数的绑定运算符
- String.prototype.at: 字符串的静态方法 at

Stage 1:

- Class and Property Decorators: Class 的修饰器
- Class Property Declarations: Class 的属性声明
- Additional export-from Statements: export 的写法改进
- String.prototype.{trimLeft,trimRight}: 字符串删除头尾空格的方法

Stage 2:

- Rest/Spread Properties: 对象的 Rest 参数和扩展运算符

Stage 3

- SIMD API: “单指令，多数据”命令集
- Async Functions: async 函数
- Object.values/Object.entries: Object 的静态方法 values()和 entries()



- String padding: 字符串长度补全
- Trailing commas in function parameter lists and calls: 函数参数的尾逗号
- Object.getPrototypeOfDescriptors: Object 的静态方法 getOwnPropertyDescriptors

Stage 4:

- Array.prototype.includes: 数组实例的 includes 方法
- Exponentiation Operator: 指数运算符

使用的时候只需要安装你想要的阶段就可以了:

```
$ npm install --save-dev babel-preset-stage-2
```

然后添加进你的 .babelrc 配置文件。

```
{
  "presets": [
    "es2015",
    "react",
+   "stage-2"
  ],
  "plugins": []
}
```